Denoising after Entropy-Based Debiasing a Robust Training Method for Dataset Bias with Noisy Labels

Sumyeong Ahn, Se-Young Yun

Kim Jaechul Graduate School of AI, KAIST, Seoul, Republic of Korea
{sumyeongahn, yunseyoung}@kaist.ac.kr

AAAI 2023

Introduction

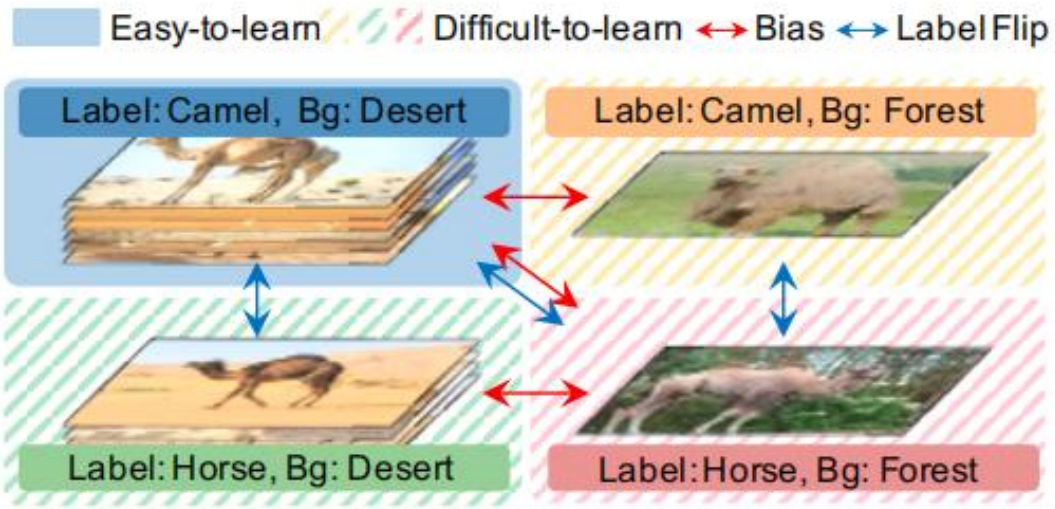


Figure 1: Examples of biased dataset with noisy labels. (1) Blue: bias-aligned, clean label samples. (2) Orange: bias-conflicting, clean labels samples. (3) Green: bias-aligned, noisy labels. (4) Red: bias-conflicting, noisy labels. The dashed background represents difficult-to-learn samples. Therefore, except for (bias-aligned, clean) case, other cases are difficult-to-learn. To mitigate dataset bias with noisy labels, training directions for each type differ. For example, (bias-aligned, noisy) case must be discarded or cleansed, while (bias-conflicting, clean) cases have to be emphasized.

- In real life, we often encounter dataset bias problems.
- This unintended bias causes the trained model to **infer erroneously** based on shortcuts (i.e., background).
- In addition to dataset bias, noisy labels are caused by many reasons and are known to degrade training mechanisms.
- Although dataset **bias and noisy labels** can occur simultaneously and independently, few studies have **addressed both problems** at once.
- The fundamental solutions of each problem are exact opposites——Difficult-to-learn samples have to be emphasized to mitigate dataset bias, while their influence should be reduced for denoising.

Introduction-DBwNL

$$\alpha = \frac{|\mathcal{D}_c|}{|\mathcal{D}|}.$$

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **Dataset Bias with Noisy Labels(DBwNL)**

$$y_{\text{given}} = \begin{cases} \tilde{y} \sim \text{Uniform}(C). & \text{with probability } \eta \\ y & \text{with probability } 1-\eta \end{cases}$$

- Dataset Bias
- When most of the samples have attributes that are **strongly correlated with the target**, we call the phenomenon dataset bias and these attributes **bias attributes**.
- We call samples whose bias attribute is **highly correlated (weakly correlated)** with the target attribute **bias-aligned (bias-conflicting)** samples.
- This dataset bias problem is **quite harmful** when the bias attribute is **easier to learn than the target attributes**, because the model loses the motivation to learn the target attribute given its sufficiently low loss.
- Noisy Labels
- We call samples whose labels are ygiven = y and ygiven ≠ y clean label and noisy label, respectively.

→ • Training a robust model on the DBwNL dataset **emphasizes bias-conflicting samples** while discarding or reducing the impact of the noisy labels.

Debiasing Meets Noisy Labels

- Relative difficulty score (`LfF` (Nam et al. 2020), `Disen` (Lee et al. 2021))

$$\mathcal{W}(x, \underline{\boldsymbol{y}_{\text{given}}}) = \frac{\mathcal{L}_{\text{CE}}(f_b(x), \underline{\boldsymbol{y}_{\text{given}}})}{\mathcal{L}_{\text{CE}}(f_b(x), \underline{\boldsymbol{y}_{\text{given}}}) + \mathcal{L}_{\text{CE}}(f_d(x), \underline{\boldsymbol{y}_{\text{given}}})}, \quad (1)$$

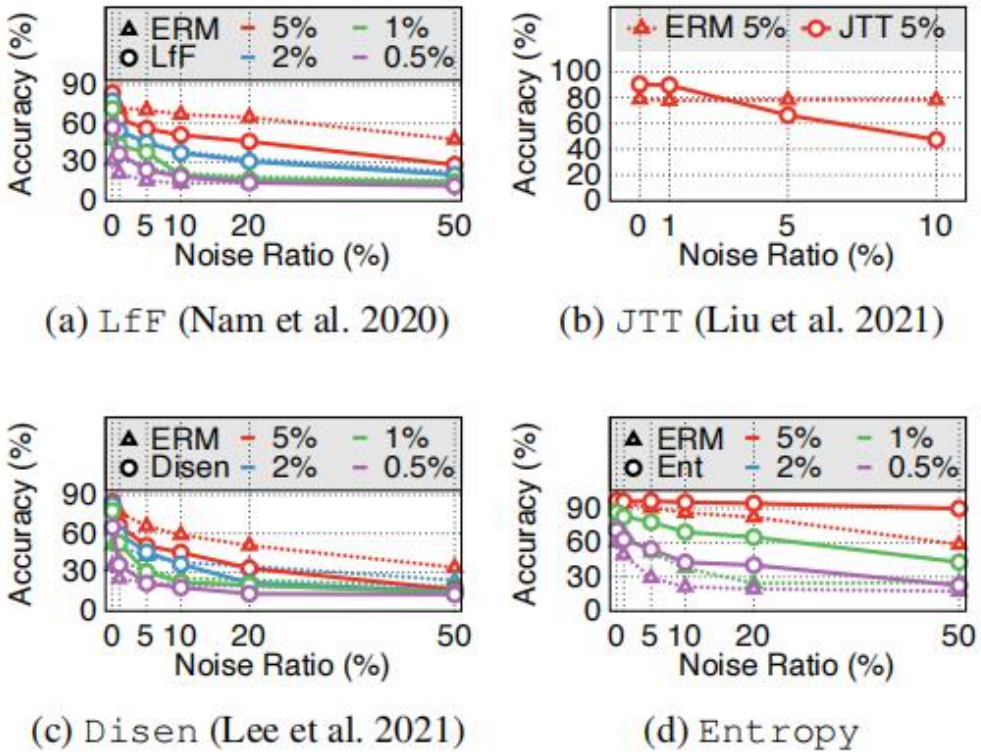
where $\mathcal{L}_{\text{CE}}$ denotes conventional cross-entropy loss and $f_b(\cdot)$ and $f_d(\cdot)$ are softmax outputs of biased and debiased models, respectively.

- Per-sample accuracy (`JTT` (Liu et al. 2021))

$$\mathcal{D}_{\text{error-set}} = \{(x, y) \text{ s.t. } \underline{\boldsymbol{y}_{\text{given}}} \neq \arg\max_c f_b(x)[c]\}, \quad (2)$$

where $f_b(x)[c]$ denotes the softmax output of logit c. The ultimate debiased model is trained on $\mathcal{D}_{\text{train}}$ composed of λ_{up} times $\mathcal{D}_{\text{error-set}}$ and the other $\mathcal{D}_{\text{corr-set}} = \mathcal{D} \setminus \mathcal{D}_{\text{error-set}}$.

- **Label-based debiasing is prone to noisy labels.**
- **Because label-based methods make incorrect emphasis.**



(a) `LfF` (Nam et al. 2020) (b) `JTT` (Liu et al. 2021)

(c) `Disen` (Lee et al. 2021) (d) `Entropy`

Figure 3: Performance when label corruption occurs. In the case of `LfF` and `Disen`, it is the unbiased test accuracy of colored MNIST, and `JTT` is the worst case test performance of the waterbird dataset. The triangle-dotted lines are the vanilla results, and the circle-solid lines represent the result of each algorithm. All algorithms except for entropy case perform worse than vanilla as noise ratio η increases.

Debiasing Meets Noisy Labels

- **Why do label-based methods suffer side-effects?**

- **Noisy labels are also emphasized when we run the labe-based algorithms.**

- **Entropy in Figure 4d, shows that the bias-conflicting and bias-aligned samples are easily distinguished regardless of whether their labels are noisy.**

- **<u>A label-free method is needed to handle DBwNL.</u>**

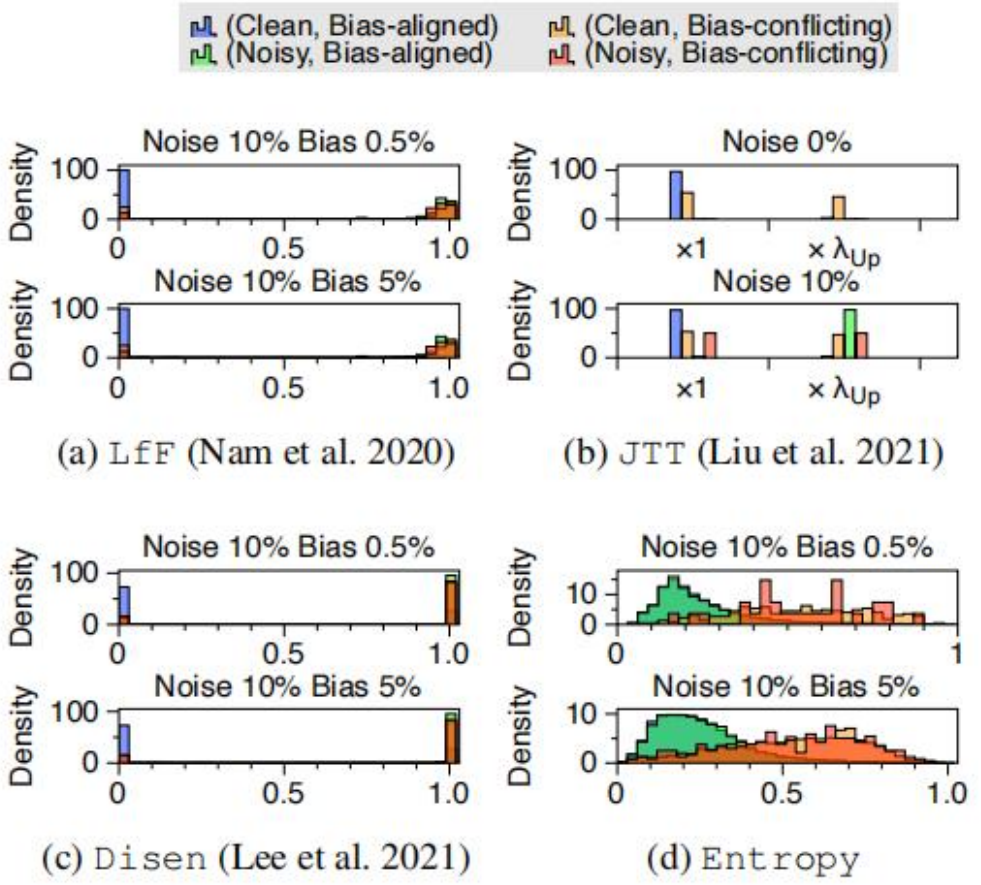
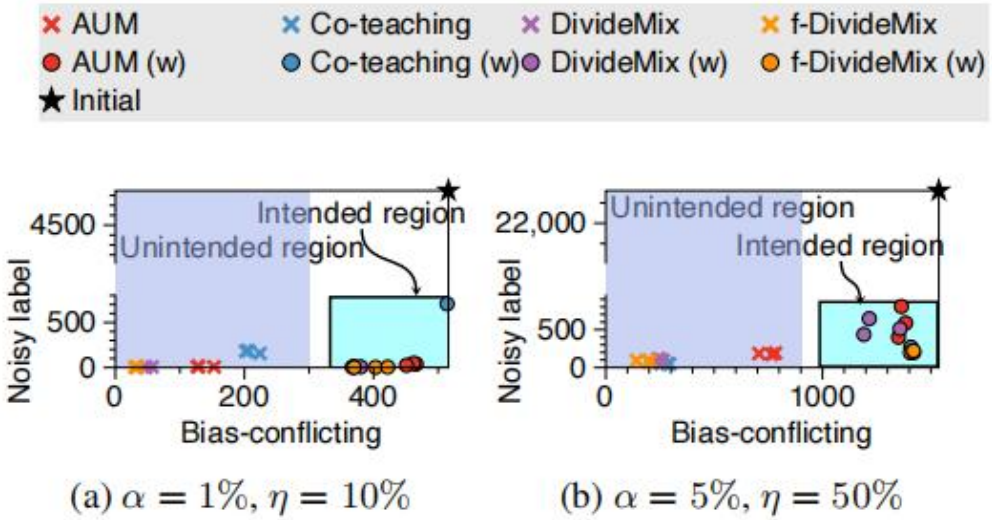


Figure 4: Score histogram of each methodology. As LfF and Disen operate online, we report the weight histogram right after the last epoch of training. Except for the Entropy case, LfF, JTT, and Disen shows entangled histograms between (noisy, aligned), (clean, conflicting), and (noisy, conflicting). By contrast, the histogram of entropy case indicates that it is clustered not according to label corruption but bias.

How Denoising Algorithms Work in the DBwNL



(a) $\alpha = 1\%, \eta = 10\%$ (b) $\alpha = 5\%, \eta = 50\%$

Figure 5: Number of remaining noisy labels and bias-conflicting samples after denoising is conducted. Star $\star$ mark represents the number of samples before cleansing, and ✗ and ● marks indicate with or without weighted training results. Since bias-conflicting samples is precious for debiasing, bias-conflicting samples have to be protected. Therefore, the region loses bias-conflicting samples (left, blue) is the unintended region. On the other hand, the region ignores noisy labels without losing the bias-conflicting samples (right, cyan) is the intended behavior.

- **Valuable bias-conflicting samples can be deemed noisy.**

- **Because the number of bias-conflicting samples is critical, removing the bias-conflicting samples prior to debiasing can cause performance degradation.**

- **Preventing bias-conflicting samples from being discarded.**
  → **denoising should be performed after highlighting bias-conflicting samples.**

Intuitions

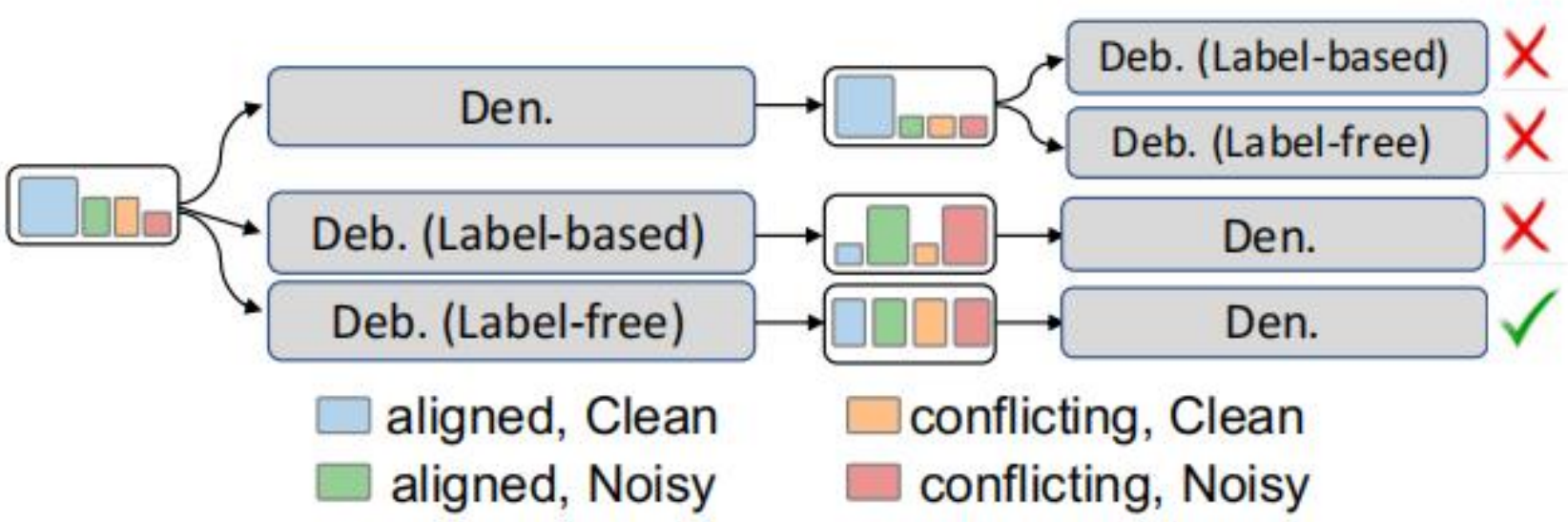


Figure 6: Case study of designing algorithm for DBwNL.

- If denoising is applied first, the bias-conflicting samples is **erased**, which is **burdensome for debiasing**
- Conversely, if debiasing is conducted first, we can choose **label-based or label-free**. If a label-based algorithm is selected, the **noisy labels are enlarged** and a burden is placed on the denoising algorithm

⇨ **(1) No label-based: label-based debiasing emphasizes noisy labels.**
(2) Debiasing before denoising: denoising algorithms should be run after debiasing emphasizes bias-conflicting samples.

Method——DENEB

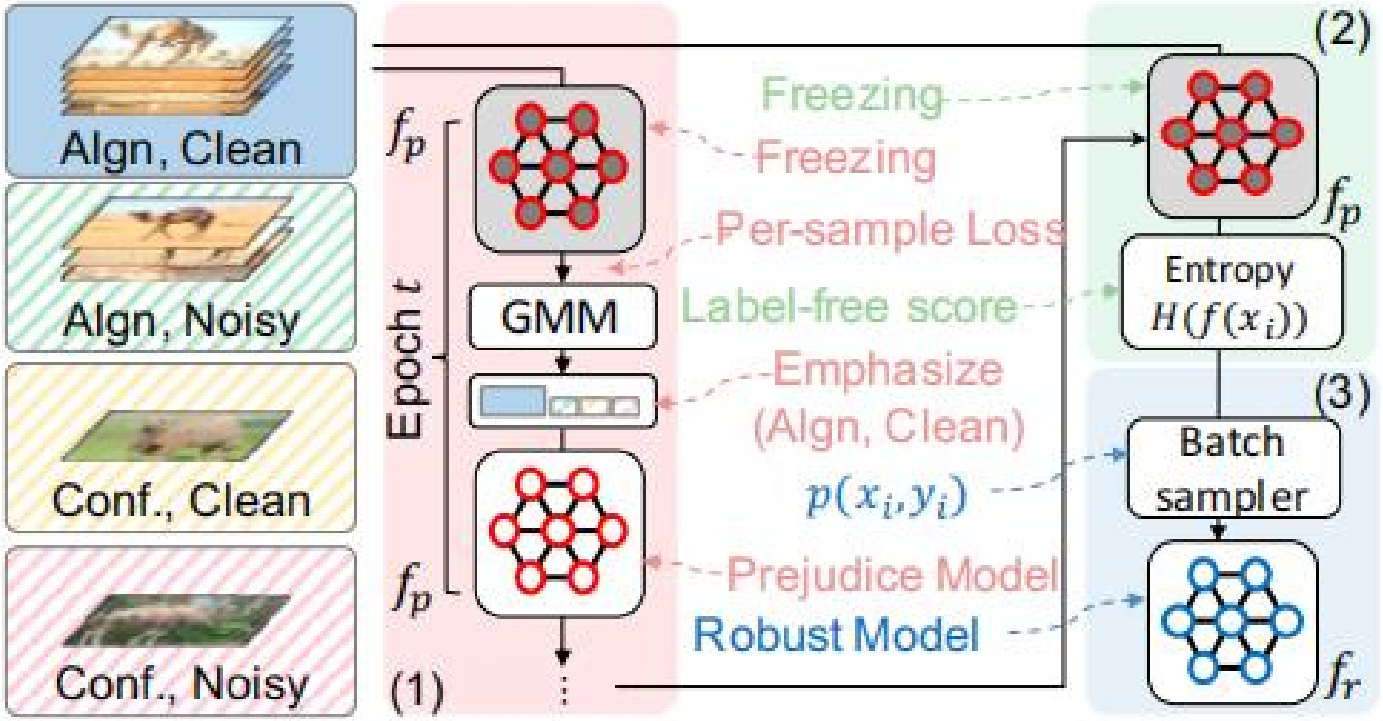


Figure 2: Overview of DENEB. It is composed of three steps. (1) Train by emphasizing (bias-aligned, clean) samples. (2) Compute label-free score, *i.e.*, entropy. (3) Train the final robust model based on the batch sampler.

Method——DENEB

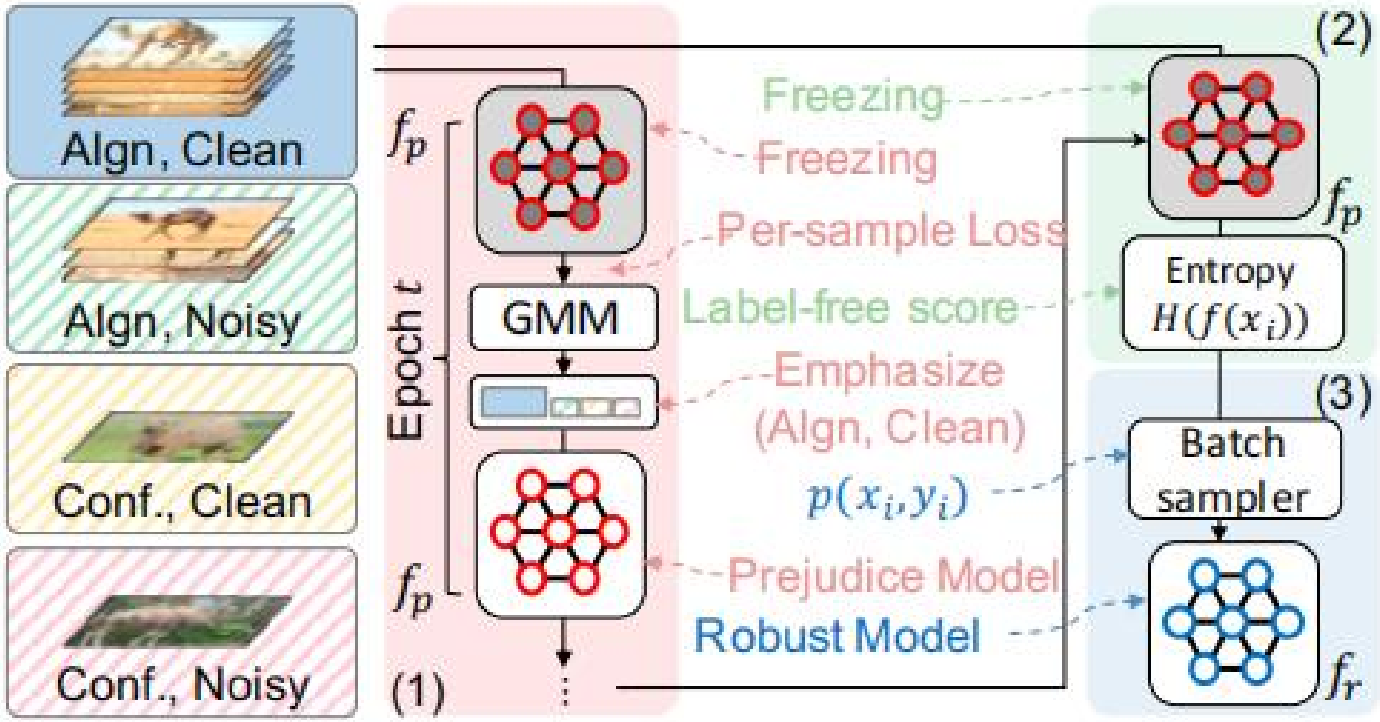


Figure 2: Overview of DENEB. It is composed of three steps. (1) Train by emphasizing (bias-aligned, clean) samples. (2) Compute label-free score, *i.e.*, entropy. (3) Train the final robust model based on the batch sampler.

Step 1: Train the prejudice model f_p

Key Aim: regardless of label corruption, the model should comprehensively learn the bias-aligned samples so that it can identify the bias conflicting samples in the next steps.

Sub step 1:

trained on $\mathcal{D}$ with conventional cross-entropy loss until the warm-up epoch e_w

Sub step 2:

$$\bar{\mathcal{D}} = \{(x_i, y_i) | g(x_i, y_i) > p_t, \text{ where } (x_i, y_i) \in \mathcal{D}\}$$

$g(x_i, y_i)$: probability of GMM

Method——DENEB

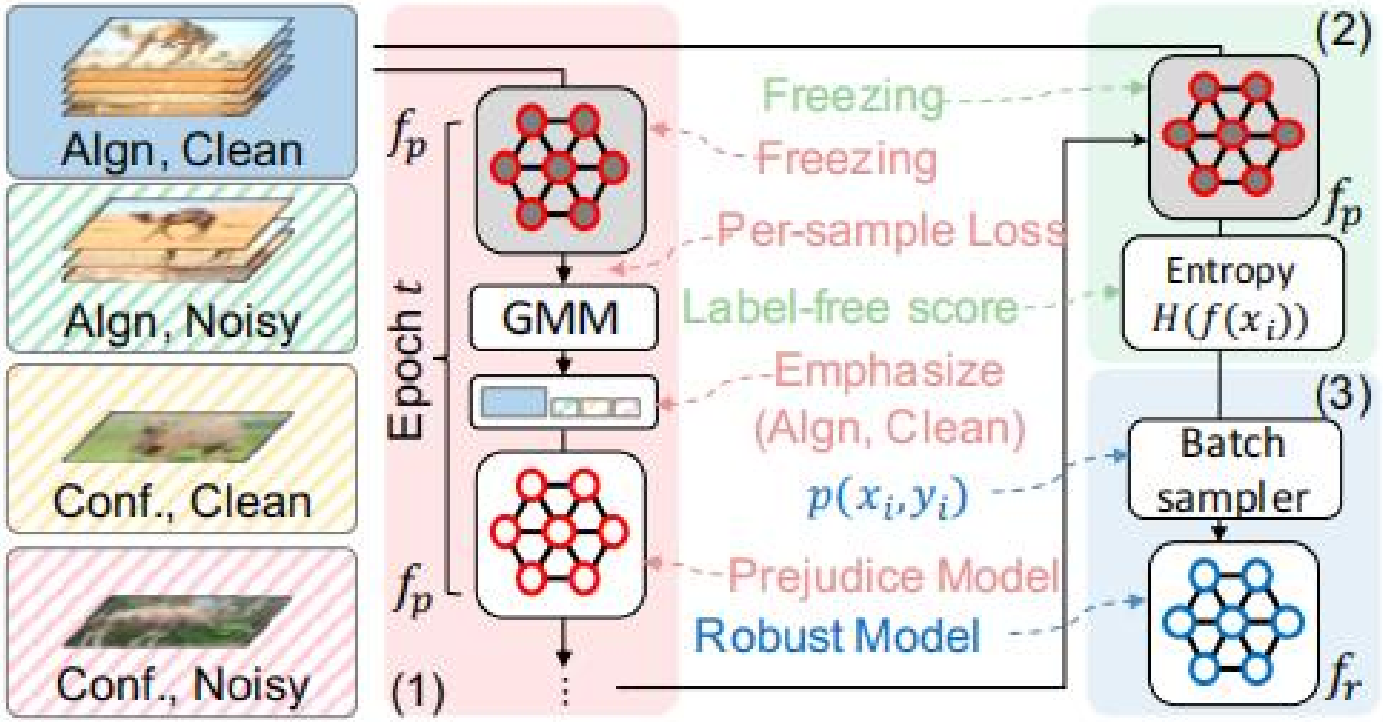


Figure 2: Overview of DENEB. It is composed of three steps. (1) Train by emphasizing (bias-aligned, clean) samples. (2) Compute label-free score, *i.e.*, entropy. (3) Train the final robust model based on the batch sampler.

Step 2: Calculate sampling probability

$$H_\tau(x) = -\sum_c^C f_p(x,\tau)[c] \times \log f_p(x,\tau)[c]$$

$$f_p(x,\tau)[j] = \frac{\exp(q_p(x)[j]/\tau)}{\sum_c \exp(q_p(x)[c]/\tau)}$$

sampling probability of each instance:

$$\mathcal{P}(x_i, y_i) = \frac{H_\tau(x_i)}{\sum_{(x_j,y_j)\in\mathcal{D}} H_\tau(x_j)}$$

→ the larger entropy samples are the bias-conflicting samples

Method——DENEB

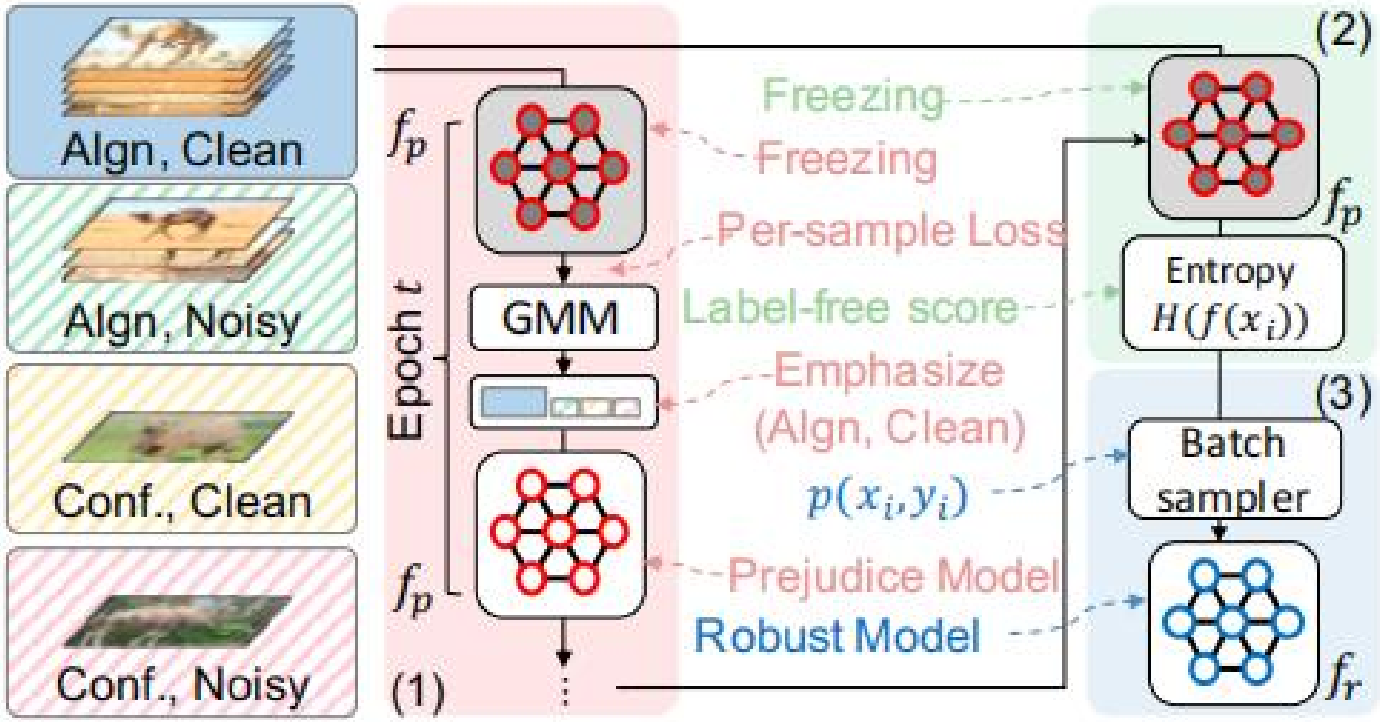


Figure 2: Overview of DENEB. It is composed of three steps. (1) Train by emphasizing (bias-aligned, clean) samples. (2) Compute label-free score, *i.e.*, entropy. (3) Train the final robust model based on the batch sampler.

Step 3: Train the robust model f_r

- Mini-batches are constructed based on the sampling probability
- The main purpose of this step is to mitigate the impact of noisy labels
- Inherit previous denoising algorithms by simply modifying the mini-batches

Experiments

Dataset	train/valid/test	#class	Target	Bias
CMNIST	54K / 6K / 10K	10	Shape	Color
CCIFAR	45K / 5K / 10K	10	Object	Blur
BAR	1,746 / 195 / 654	6	Action	Place
BFFHQ	17,280/1,920/1,000	2	Gender	Age

Table 1: Benchmark Summary

Algorithm	Colored MNIST		Corrupted CIFAR-10	
	$\alpha = 1\%, \eta = 10\%$	$\alpha = 5\%, \eta = 50\%$	$\alpha = 1\%, \eta = 10\%$	$\alpha = 5\%, \eta = 50\%$
Vanilla	39.24 ± 1.91%	70.13% ± 3.42 %	25.43% ± 0.84 %	31.86% ± 0.96 %
Debiasing				
LfF	29.87 ± 1.36%	57.97% ± 1.79 %	24.51% ± 1.30 %	29.68% ± 2.63 %
JTT	63.24 ± 2.60%	77.16% ± 1.15 %	23.75% ± 0.61 %	24.52% ± 0.98 %
EIIL	24.53 ± 0.31%	42.25% ± 1.43 %	20.30% ± 1.08 %	22.66% ± 1.94 %
Disen	31.49 ± 5.44%	69.20% ± 4.13 %	22.52% ± 0.38 %	28.35% ± 4.49 %
Denoising				
GCE	19.52 ± 1.98%	73.45% ± 7.62 %	24.96% ± 1.53 %	30.72% ± 0.74 %
SCE	30.95 ± 2.87%	62.10% ± 5.02 %	23.34% ± 1.73 %	29.87% ± 1.00 %
ELR+	24.76 ± 0.90%	49.38% ± 3.74 %	22.10% ± 0.37 %	30.84% ± 0.43 %
AUM	23.89 ± 2.60%	49.51% ± 6.62 %	23.55% ± 1.10 %	28.06% ± 2.38 %
Co-teaching	41.89 ± 1.45%	76.64% ± 5.52 %	25.14% ± 0.27 %	26.84% ± 0.52 %
DivideMix	20.48 ± 1.94%	33.66% ± 2.91 %	18.86% ± 0.28 %	22.03% ± 0.59 %
f-DivideMix	22.06 ± 1.70%	39.92% ± 3.26 %	19.67% ± 0.25 %	27.60% ± 0.54 %
DENEB				
DENEB	**91.81 ± 0.84%**	**94.55% ± 0.22 %**	**26.05% ± 0.54 %**	**35.32% ± 1.03 %**

Table 2: Unbiased test accuracy on CMNIST and CCIFAR. Best-performing results are marked in bold. All results are averaged on three independent runs. DENEB represents *i.e.*, $\mathcal{A}_{\text{den}} = \texttt{GCE}$.

Experiments

Algorithm	BAR $\eta = 10\%$	BFFHQ $\eta = 10\%$
Vanilla	54.37 $\pm$ 1.10%	71.38 $\pm$ 0.58%
LfF	53.62 $\pm$ 1.81%	54.35 $\pm$ 0.91%
JTT	55.67 $\pm$ 2.16%	70.18 $\pm$ 1.47%
Disen	55.80 $\pm$ 3.05%	67.44 $\pm$ 2.57%
GCE	56.39 $\pm$ 0.95%	68.45 $\pm$ 2.98%
Co-teaching	54.99 $\pm$ 1.28%	69.28 $\pm$ 1.24%
DivideMix	52.01 $\pm$ 1.51%	72.20 $\pm$ 0.58%
DENEB	**62.30 $\pm$ 0.91%**	**75.24 $\pm$ 0.68%**

Table 3: Unbiased test accuracy on BAR and BFFHQ. Best performing results are marked in bold. All results are averaged on three independent runs.

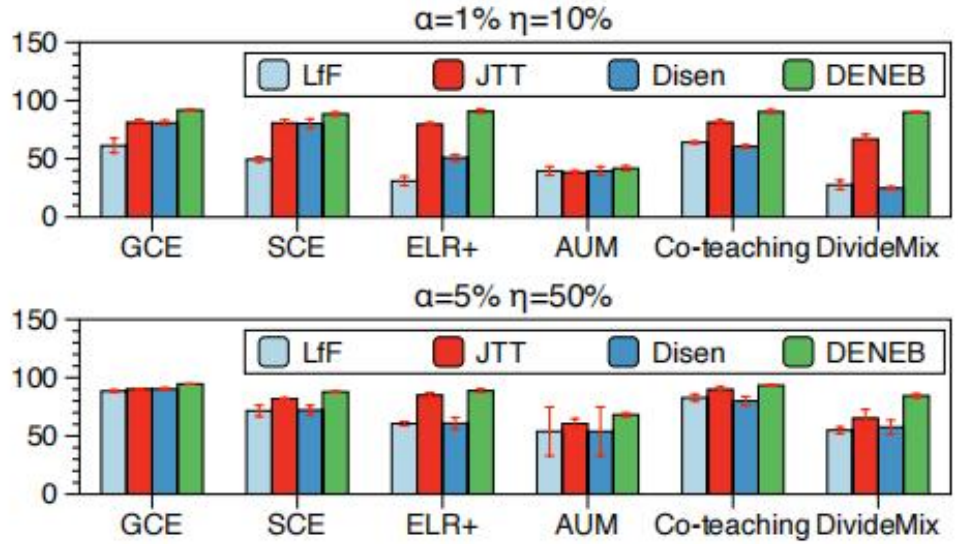


Figure 7: Combination result of Colored MNIST benchmark. All cases are the performances of Debiasing $\rightarrow$ Denoising, *i.e.*, obtain per-sample weights from DENEBand then run GCE for DENEB$\rightarrow$GCE case.

Thanks